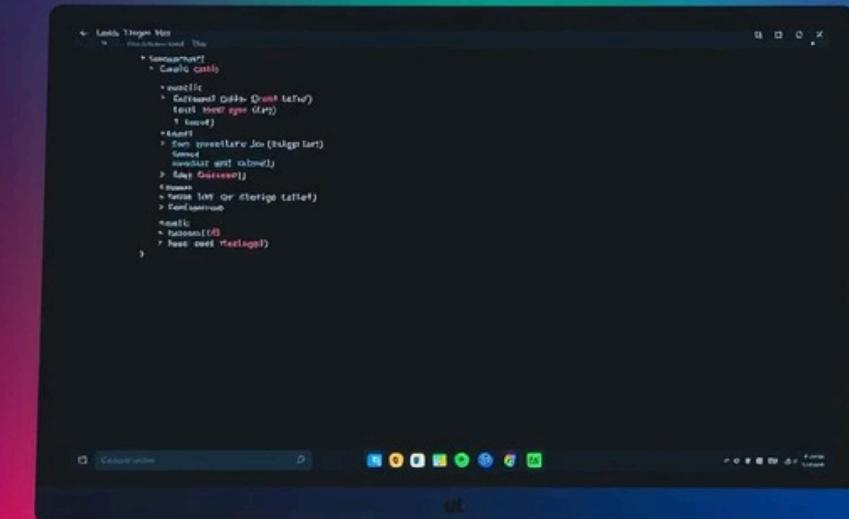


Introduction to Linux

Linux is an open-source operating system based on Unix architecture. It is widely recognized for its flexibility and robustness. The core of Linux is the kernel, which manages hardware resources, while the operating system includes additional software that provides user interfaces and tools.

Unlike proprietary systems, Linux's open-source nature allows anyone to view, modify, and distribute its code. This has fostered a large community and a rich ecosystem of software and distributions.





History of Linux

Unix Origins

Developed in 1969 at AT&T Bell Labs, Unix laid the foundation for modern operating systems.

GNU Project

Richard Stallman's Free Software Foundation and GNU Project contributed essential tools for Linux.

1

2

3

Linux Kernel

Linus Torvalds created the Linux kernel in 1991, initiating the Linux project.

Linux Distributions (Distros)

Popular Distros

- Ubuntu
- Fedora
- CentOS
- Red Hat
- Kali

Use Cases & Package Managers

- Desktop, Server, Security, Embedded
- Package managers: apt, yum, dnf, zypper

Linux Architecture

Kernel

Core component managing hardware and system resources.

Shell

User interface for command execution and scripting.

File System

Organizes data storage and access.

User vs Kernel Space

Separation between user applications and core system functions.



Features of Linux



Multitasking & Multiuser

Supports running multiple processes and users simultaneously.



Portability

Runs on diverse hardware platforms from servers to embedded devices.



Security & Stability

Robust protection mechanisms and reliable performance.



Open-source Ecosystem

Vibrant community contributing to continuous development.

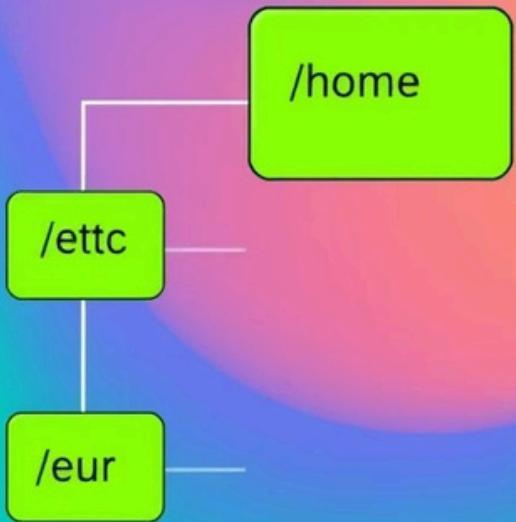
Linux Command Line (CLI)

Shells

- bash
- sh
- zsh

Common Commands

- ls, cd, pwd
- mkdir, rm, mv, cp
- Case sensitivity, tab completion, piping



Linux Filesystem Hierarchy

Root Directory /

Top-level directory containing all files and folders.

Key Directories

- /home - user files
- /etc - configuration files
- /var - variable data
- /usr, /bin, /lib, /tmp

Files vs Directories

Organized structure with mounting of devices supported.

Linux Users and Permissions

User Types

- Root (superuser)
- Regular users
- System users

Permissions & Ownership

- File ownership chmod, chown,
- chgrp commands Read, write,
- execute permissions

Linux vs Windows - Introduction



Over view

Linux and Windows are two major operating systems with distinct architectures and philosophies.



Market Comparison

Windows dominates desktop markets, while Linux is prevalent in servers and specialized uses.



Target Audiences

Windows targets general consumers; Linux appeals to developers, enterprises, and enthusiasts.



File System Differences: Linux vs Windows

Windows File Systems

- NTFS FAT32
- exFAT Path
- format: C:\
-

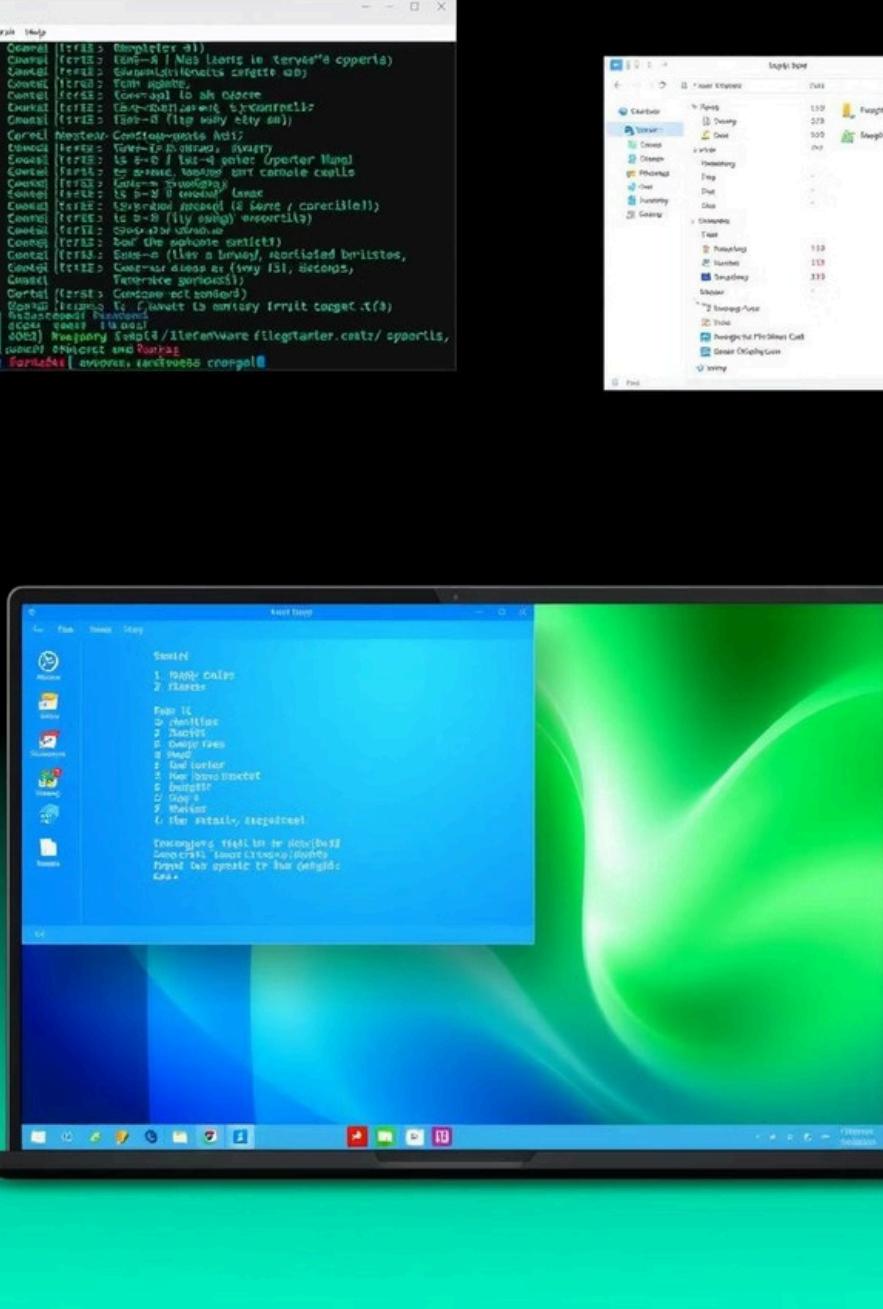
Linux File Systems

- ext3, ext4
- Btrfs, XFS
- Path format: /
- Case sensitive

Linux vs Windows: Key Differences and Use Cases

This presentation explores the fundamental differences between Linux and Windows operating systems. We will cover command line versus graphical interfaces, software management, security models, performance, and stability. Finally, we'll discuss practical use cases for each system, helping you understand when to choose Linux or Windows based on your needs.





Command Line Interface vs Graphical User Interface

Linux Preference for CLI

Linux users often favor the command line interface (CLI) for its power and flexibility, enabling advanced control and scripting capabilities.

Windows GUI Dependency

Windows relies heavily on its graphical user interface (GUI), making it more accessible for casual users but less suited for automation.

Scripting and Automation

Linux's CLI excels in scripting and automation, streamlining repetitive tasks and system management efficiently.

Software Management Approaches

Windows Software Installation

Windows primarily uses .exe and .msi installers along with the Microsoft Store for software distribution.

This approach is user-friendly but can lead to fragmented dependency management.

Linux Package Managers

Linux uses package managers like apt and dnf to handle software installation and updates.

These tools manage dependencies automatically, ensuring system stability and consistency.

Security Models Compared

Linux Security

- File permissions and ownership
- SELinux for mandatory access control
- iptables for firewall management

Windows Security

- Active Directory for user management
- User Account Control (UAC) for privilege elevation
- Windows Defender for malware protection

Vulnerability Handling

Both systems provide regular updates and patches, but Linux's open-source nature allows faster community-driven fixes.



Performance and Stability

Linux Performance

Linux is lightweight and optimized for server environments, requiring fewer reboots and offering high stability.

Windows Performance

Windows prioritizes user-friendliness but tends to consume more resources and may require more frequent restarts.

When to Use Linux

Server Environments

Linux is preferred for servers due to its stability, security, and resource efficiency.

Developers and Administrators

Its powerful CLI and scripting capabilities make Linux ideal for developers and system admins.

Open Source Enthusiasts

Users who value customization and transparency often choose Linux.





When to Use Windows

Desktop Environments

Windows is widely used for personal computers due to its intuitive GUI and extensive software compatibility.

Casual Users

Its ease of use makes it suitable for everyday tasks and non-technical users.

Enterprise Integration

Windows integrates well with enterprise tools like Active Directory and Microsoft Office.



Summary and Coexistence



Choosing Based on Needs

Select Linux or Windows depending on your environment, user type, and technical requirements.



Server vs Desktop

Linux excels in servers and technical roles, while Windows dominates desktop and casual use.



Coexistence Options

Dual-boot setups and virtualization allow users to leverage the strengths of both systems.